



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Journal of Computational and Applied Mathematics 180 (2005) 443–459

JOURNAL OF  
COMPUTATIONAL AND  
APPLIED MATHEMATICS[www.elsevier.com/locate/cam](http://www.elsevier.com/locate/cam)

# On the degree elevation of Bernstein polynomial representation

Bolesław Szafnicki\*

*Dr. Szafnicki, CAD Applications, Scharrkoppelweg 22, D-38557 Osloss, Germany*

Received 21 April 2004; received in revised form 21 September 2004

## Abstract

The polynomials determined in the Bernstein (Bézier) basis enjoy considerable popularity in computer-aided design (CAD) applications. The common situation in these applications is, that polynomials given in the basis of degree  $n$  have to be represented in the basis of higher degree. The corresponding transformation algorithms are called algorithms for degree elevation of Bernstein polynomial representations. These algorithms are only then of practical importance if they do not require the ill-conditioned conversion between the Bernstein and the power basis. We discuss all the algorithms of this kind known in the literature and compare them to the new ones we establish. Some among the latter are better conditioned and not more expensive than the currently used ones. All these algorithms can be applied componentwise to vector-valued polynomial Bézier representations of curves or surfaces.

© 2004 Elsevier B.V. All rights reserved.

MSC: 65D17

Keywords: Degree elevation; Bernstein basis

## 1. Introduction

The common situation in *computer-aided design (CAD)* applications is that the polynomial

$$p(t) = \sum_{i=0}^n x_i B_i^n(t), \quad x_i \in \mathbb{R}, \quad t \in [0, 1],$$

\* Tel.: +49 5362 722 32; fax: +49 5361 186 995.

E-mail address: [boleslaw.szafnicki@gedas.de](mailto:boleslaw.szafnicki@gedas.de) (B. Szafnicki).

which is given in the *Bernstein (Bézier) basis*  $\{B_i^n(t)\}_{i=0}^n = \{\binom{n}{i}(1-t)^{n-i}t^i\}_{i=0}^n$  of degree  $n$ , is to be represented in the form

$$p(t) = \sum_{i=0}^{n+k} y_i B_i^{n+k}(t), \quad y_i \in \mathbb{R}, \quad t \in [0, 1],$$

that is, in the Bernstein basis of degree  $n+k$ ,  $k \geq 1$  (see [1–6,8–10,12,14–18,21–25,28–33,36,38,39,41]).

This situation may be caused by the following reasons.

- (a) The addition, subtraction and division for polynomials in Bernstein form necessitate, in general, such *degree elevation* [18, Sections 4.1, 4.3], [2, Sections 3.2, 3.4].
- (b) The use of the Bernstein (Bézier) basis in *CAD systems* creates a well-known geometric connection (see, e.g., [5, Sections 6–8], [16, Sections 3.2–3.3, 5.2]) between the shape of the curve  $\mathbf{c}$  defined by  $p(t)$  and its *control polygon*, that is, the piecewise linear curve connecting the *control points* of the curve  $\mathbf{c}$ , where the control points are the points that represent the coefficients  $x_i$  on the screen. The CAD users apply extensively this connection to model the curves interactively by manipulating their control points. The higher the degree  $n$  of the used Bernstein basis, the closer is this geometric connection [6,10,22–24,34], and the wider is the range of possible shape modifications of the curve.
- (c) In a CAD system a set of curves is often used to create a *composite curve* or *surface*. It usually can only be done if the Bernstein bases, these curves are represented in, are of the same degree. This and the transfer of curves represented in the Bernstein basis of degree  $n$  to a CAD system, which uses the basis of degree  $n+k$ , are two other reasons for degree elevation.
- (d) The *condition number* for a root of the polynomial  $p(t)$  decreases monotonically under degree elevation of the Bernstein basis, the polynomial is represented in [17, Theorem 4, Table 3]. This condition number measures the sensitivity of the root to random perturbations in coefficients  $x_i$  [40, Chapter 2, Section 7].

For the reasons mentioned in (a)–(d) we need the algorithms for the degree elevation of Bernstein polynomial representation which are accurate and fast.

Because the conversion between the Bernstein and the *power basis*  $\{t^i\}_{i=0}^n$  is *ill conditioned* [7,20], it is of particular importance, that this conversion is not used in the degree elevation algorithms. We describe a set of such current and new algorithms in Section 2 and we show, in Section 3, that the latter are not more *expensive* than the former.

Applying the results obtained in Section 4 for the *conditioning of degree elevation matrices*, we study in Section 5 the accuracy of our algorithms and we show that some current can be replaced by such new ones that are substantially *better conditioned* and thus, in general, substantially more accurate.

The polynomials considered here are *real-valued* ( $x_i \in \mathbb{R}$ ). In *computer-aided geometric design* (CAGD) are usually used (see, e.g., [5, Section 35; 16, Section 16.5]) the so-called *Bézier curves* determined by

$$\mathbf{p}(t) = \sum_{i=0}^n \mathbf{x}_i B_i^n(t), \quad \mathbf{x}_i \in \mathbb{R}^N, \quad t \in [0, 1], \quad N = 2, 3 \quad (1)$$

and rectangular Bézier surfaces or patches determined by

$$\mathbf{s}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{x}_{i,j} B_i^m(u) B_j^n(v), \quad \mathbf{x}_{i,j} \in \mathbb{R}^3, \quad u, v \in [0, 1]. \quad (2)$$

To compute the coefficients  $\mathbf{y}_i \in \mathbb{R}^N$  in the Bernstein basis of degree  $n + k$  of the *vector-valued polynomial* given by (1), that is, to elevate to  $n + k$  the degree of the basis this polynomial is represented in, one applies a degree elevation algorithm to each component of the vectors  $\mathbf{x}_i$  separately; in other words, the computation is carried out *componentwise*. On the other hand, the degree elevation of the *tensor product Bézier representation of surfaces* (2) consists in applying several times a degree elevation algorithm for the curves (see, e.g., [16, Chapter 16]). Hence the facts stated in Sections 3 and 5 are also valid for the curves and surfaces defined in form (1) or (2), respectively.

The degree elevation algorithms for the *triangular Bézier patches* have a specific character (see, e.g., [11, Section 2.1.8; 13, Section 1.6; 16, Section 18.8; 35, Section 3]) and will not be treated in this paper. We conclude with Section 6.

## 2. Degree elevation algorithms

To shorten our statements we first set up an appropriate notation (cf. [37]).

**Definition 2.1.** Let  $\mathbf{X}^n = \{x_i\}_{i=0}^n \in \mathbb{R}^{n+1}$ . We define

$$\downarrow^m \mathbf{X}^n = \{x_i\}_{i=0}^m, \quad \uparrow^m \mathbf{X}^n = \{x_{n-i}\}_{i=0}^m, \quad m = -1, \dots, n,$$

where  $m = -1$  determines the empty sequence. We treat  $\mathbf{X}^n$  as  $\downarrow^n \mathbf{X}^n$  and alternatively interpret sequences as vectors with the standard operations. Given the sequence  $\mathbf{X}^n$ , we define the polynomial in the Bernstein basis of degree  $n$  by

$$\beta(\mathbf{X}^n, t) = \sum_{i=0}^n x_i B_i^n(t) = \sum_{i=0}^n x_i \binom{n}{i} (1-t)^{n-i} t^i, \quad t \in [0, 1].$$

The following theorem enables us to define the current and new algorithms for degree elevation of Bernstein polynomial representation and discuss their conditioning in a uniform way.

**Theorem 2.2.** Let  $\mathbf{X}^n \in \mathbb{R}^{n+1}$ ,  $\mathbf{Y}^{n+k} \in \mathbb{R}^{n+k+1}$ ,  $k, n \in \mathbb{N}$  and let the matrices

$$\boldsymbol{\Omega}_{\mu}^{n,k} = \{\omega_{i,j}^{n,k}\}_{i,j=0}^{\mu, \min(n,\mu)}, \quad \mu = -1, \dots, n+k, \quad (3)$$

where  $\mu = -1$  determines the empty matrix, be defined by

$$\omega_{i,j}^{n,k} = \begin{cases} \binom{k}{i-j} \binom{n}{j} / \binom{n+k}{i}, & \max(0, i-k) \leq j \leq \min(n, i), \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Then

$$\beta(\mathbf{X}^n, t) = \beta(\mathbf{Y}^{n+k}, t) \quad \text{for all } t \in [0, 1] \quad (5)$$

if and only if for all  $\mu, \nu = 0, \dots, n+k$

$$\downarrow^\mu \mathbf{Y}^{n+k} = \Omega_\mu^{n,k} [\downarrow^{\min(n,\mu)} \mathbf{X}^n], \quad \uparrow^\nu \mathbf{Y}^{n+k} = \Omega_\nu^{n,k} [\uparrow^{\min(n,\nu)} \mathbf{X}^n]. \quad (6)$$

**Proof.** For all  $t \in [0, 1]$  it holds  $\sum_{l=0}^k B_l^k(t) = 1$  (see, e.g., [16, (4.5)]), thus

$$\beta(\mathbf{X}^n, t) = \sum_{l=0}^k B_l^k(t) \beta(\mathbf{X}^n, t) = \sum_{i=0}^{n+k} \sum_{j=\max(0, i-k)}^{\min(n, i)} x_j B_j^n(t) B_{i-j}^k(t).$$

For all  $j = \max(0, i-k), \dots, \min(n, i)$ ,  $i = 0, \dots, n+k$ ,  $t \in [0, 1]$  we have

$$B_j^n(t) B_{i-j}^k(t) = \binom{n+k}{i}^{-1} \binom{k}{i-j} \binom{n}{j} B_i^{n+k}(t)$$

and  $\{B_i^{n+k}(t)\}_{i=0}^{n+k}$  is a basis in the collection of polynomials of degree at most  $n+k$ . Therefore (5) takes place only if for all  $\mu, \nu = 0, \dots, n+k$

$$\downarrow^\mu \mathbf{Y}^{n+k} = \left\{ \binom{n+k}{i}^{-1} \sum_{j=\max(0, i-k)}^{\min(n, i)} \binom{k}{i-j} \binom{n}{j} x_j \right\}_{i=0}^\mu, \quad (7)$$

$$\uparrow^\nu \mathbf{Y}^{n+k} = \left\{ \binom{n+k}{i}^{-1} \sum_{j=\max(0, i-k)}^{\min(n, i)} \binom{k}{i-j} \binom{n}{j} x_{n-j} \right\}_{i=0}^\nu, \quad (8)$$

which together with (3), (4) completes the proof.  $\square$

For  $k = 1$  and  $\mu, \nu = 0, \dots, n+1$ , we will use the following simplifications of (7), (8) obtained by manipulating their binomial coefficients

$$\downarrow^\mu \mathbf{Y}^{n+1} = \{[ix_{i-1} + (n+1-i)x_i]/(n+1)\}_{i=0}^\mu, \quad (9)$$

$$\uparrow^\nu \mathbf{Y}^{n+1} = \{[ix_{n-i+1} + (n+1-i)x_{n-i}]/(n+1)\}_{i=0}^\nu. \quad (10)$$

A quite different way of computing  $\mathbf{Y}^{n+k}$ , than that described by (7)–(8), was given by Trump and Prautzsch in [39]. They proved, using (3) and the first part of (6) with  $\mu = n+k$ , that

$$\mathbf{Y}^{n+k} = \{b_i^{\min(k, i)}\}_{i=0}^{n+k}, \quad (11)$$

where

$$b_i^0 = x_i, \quad i = 0, \dots, n, \quad (12)$$

$$b_{s+j}^s = \begin{cases} \frac{(j+k-s+1)b_{s+j-1}^{s-1} + (n-j)b_{s+j}^{s-1}}{n+k-s+1}, & j = 0, \dots, n-1, \\ x_n, & j = n, \end{cases} \quad s = 1, \dots, k. \quad (13)$$

We note that for any fixed  $\mu = -1, \dots, n+k$  and for  $\nu = n+k-\mu-1$ , the vector  $\mathbf{Y}^{n+k}$  is uniquely determined by (6). This gives rise to the following definition.

**Definition 2.3.** Assume that for every  $\mathbf{X}^n \in \mathbb{R}^{n+1}$  and  $k \geq 1$  an algorithm  $\mathbf{a}$  yields  $\mathbf{Y}^{n+k}$  determined by (6) with a fixed  $\mu = -1, \dots, n+k$  and  $v = n+k - \mu - 1$ . Then  $\mathbf{a}$  is said to be an *algorithm for the  $k$ -fold degree elevation of Bernstein polynomial representation* and is designated by  $\mathbf{a}_\mu^{n,k}$ .

Using Definition 2.3 and relations (7)–(13), we describe the algorithms  $\mathbf{s}_\mu^{n,k}$ , where  $\mu = -1, \dots, n+k$ , and the algorithm  $\mathbf{t}_{n+k}^{n,k}$  as follows:

#### Algorithms 2.4.

$\mathbf{s}_\mu^{n,k}$ :  $\downarrow^\mu \mathbf{Y}^{n+k}, \uparrow^{n+k-\mu-1} \mathbf{Y}^{n+k}$  are computed for  $k \neq 1$  by (7)–(8) and for  $k = 1$  by (9)–(10),  
 $\mathbf{t}_{n+k}^{n,k}$ :  $\mathbf{Y}^{n+k}$  is computed by (11), with  $\{b_i^{\min(k,i)}\}_{i=0}^{n+k}$  given by (12)–(13).

For  $k > 1$ , these algorithms serve us in turn to describe two other ones.

#### Algorithms 2.5.

$\bar{\mathbf{s}}_{n+k}^{n,k}$ :  $\mathbf{Y}^{l+1} = \mathbf{s}_{l+1}^{l,1}(\mathbf{Y}^l)$  is computed for  $l = n, \dots, n+k-1$  by (9)–(10),  
 $\bar{\mathbf{t}}_{n+k}^{n,k}$ :  $\mathbf{Y}^{l+1} = \mathbf{t}_{l+1}^{l,1}(\mathbf{Y}^l)$  is computed for  $l = n, \dots, n+k-1$  by (11)–(13),

where  $\mathbf{Y}^n = \mathbf{X}^n$ ,  $\mathbf{a}(\mathbf{Y}^l)$  denotes the result of applying the algorithm  $\mathbf{a}$  to  $\mathbf{Y}^l$ , and the algorithms  $\mathbf{s}_{l+1}^{l,1}$ ,  $\mathbf{t}_{l+1}^{l,1}$  are defined in Algorithms 2.4.

The current algorithms for degree elevation of Bernstein polynomial representation are  $\mathbf{s}_{n+k}^{n,k}$ ,  $\bar{\mathbf{s}}_{n+k}^{n,k}$  and  $\mathbf{t}_{n+k}^{n,k}$  (see, e.g., [5,18,39]). Other algorithms described in Algorithms 2.4 and 2.5 do not seem to be well known. We show in the next section that these new algorithms are not more expensive and, in Section 5, that some of them are substantially better conditioned than the current ones.

### 3. Costs of algorithms

We determine now the costs of our algorithms. We do this assuming that the binomial coefficients are stored, the multiplications and divisions by the factor 1 are not carried out, and that the computation of the two equal integer factors multiplying  $x_j$  and  $x_{n-j}$  in (7), (8) is performed only once.

The costs of Algorithms 2.4 are listed in Table 1 and those of Algorithms 2.5 in Table 2.

From Tables 1 and 2 follows easily that the algorithm  $\bar{\mathbf{s}}_{n+k}^{n,k}$  has the *complexity of order*  $O(k^2)$  and is more expensive than the algorithm  $\mathbf{t}_{n+k}^{n,k}$  which has the complexity of order  $O(k)$ . This fact was observed in [39, Section 2].

However, we can easily state that the algorithm  $\mathbf{s}_\mu^{n,k}$ , with an arbitrary  $\mu = -1, \dots, n+k$ , is for  $k > 1$  less expensive than the algorithm  $\mathbf{t}_{n+k}^{n,k}$ .

Finally, using Tables 1 and 2, we see that for an arbitrary  $\mu = -1, \dots, n+1$  the costs of floating operations for the algorithms  $\mathbf{t}_{n+1}^{n,1}$ ,  $\mathbf{s}_\mu^{n,1}$  are the same and so are these costs for the algorithms  $\bar{\mathbf{t}}_{n+k}^{n,k}$ ,  $\bar{\mathbf{s}}_{n+k}^{n,k}$ .

Table 1 shows that the costs of the algorithm  $\mathbf{s}_\mu^{n,k}$  do not depend on  $\mu$ . On the contrary, the conditioning of any algorithm  $\mathbf{a}_\mu^{n,k}$  depends only on  $\mu$ . We prove this in Section 5 using the results of the next section.

Table 1

Operation counts for the algorithms  $\mathbf{s}_\mu^{n,k}$ , with an arbitrary  $\mu = -1, \dots, n+k$ , and  $\mathbf{t}_{n+k}^{n,k}$ .  $I^{n,k}$  is defined by  $I^{n,k} = [n - (-1)^{n+k}](k+1)/2$

Algorithm	$\pm$	Floating operations		Integer operations	
		$\times$	$/$	$\pm$	$\times$
$\mathbf{s}_\mu^{n,k}$	$kn$	$kn + n + k - 1$	$n + k - 1$	—	$I^{n,k}$
$\mathbf{t}_{n+k}^{n,k}$	$kn$	$2kn$	$kn$	$2(n+1)k$	—

Table 2

Operation counts for the algorithms  $\bar{\mathbf{s}}_{n+k}^{n,k}$ ,  $\bar{\mathbf{t}}_{n+k}^{n,k}$ , where  $k > 1$ .  $K^{n,k}$  is defined by  $K^{n,k} = (-1)^n[1 - (-1)^{k+1}]/2$

Algorithm	$\pm$	Floating operations		Integer operations	
		$\times$	$/$	$\pm$	$\times$
$\bar{\mathbf{s}}_{n+k}^{n,k}$	$kn + \binom{k}{2}$	$2[kn + \binom{k}{2}]$	$kn + \binom{k}{2}$	—	$kn + \binom{k}{2} + K^{n,k}$
$\bar{\mathbf{t}}_{n+k}^{n,k}$	$kn + \binom{k}{2}$	$2[kn + \binom{k}{2}]$	$kn + \binom{k}{2}$	$2[kn + \binom{k+1}{2}]$	—

#### 4. Conditioning of degree elevation matrices

An algorithm  $\mathbf{a}_\mu^{n,k}$  yields the vectors

$$\downarrow^\mu \mathbf{Y}^{n+k}, \quad \uparrow^{n+k-\mu-1} \mathbf{Y}^{n+k} \quad (14)$$

determined by the linear transformations (6) with the matrices

$$\mathbf{\Omega}_\mu^{n,k}, \quad \mathbf{\Omega}_{n+k-\mu-1}^{n,k},$$

respectively. We will now examine the conditioning of these matrices.

**Proposition 4.1.** *If  $\mu \in [k-1, n]$  then the matrices  $\mathbf{\Omega}_\mu^{n,k}$ ,  $\mathbf{\Omega}_{n+k-\mu-1}^{n,k}$  are nonsingular. If  $\mu \notin [k-1, n]$  then at least one of the numbers  $\mu, n+k-\mu-1$  determining the dimensions of these matrices is greater than  $n$ .*

**Proof.** If  $\mu \in [k-1, n]$  then it is easily verified that  $k-1 \leq n+k-\mu-1 \leq n$ , which together with (3)–(4) gives the first part of the hypothesis. For its second part, it suffices to consider  $\mu \leq n$ . Then  $k-1 > n$  and  $n+k-\mu-1 \geq k-1 > n$  or, since  $\mu \notin [k-1, n]$ , we have  $\mu < k-1 \leq n$  and thus  $n+k-\mu-1 > n$ .  $\square$

The conditioning of a linear transformation whose matrix is nonsingular is usually measured by use of the *matrix condition numbers*.

The condition number  $\kappa_p(\mathbf{A})$ , associated with the vector norm  $L_p(p \geq 1)$ , of a nonsingular matrix  $\mathbf{A}$  is defined [27, Section 6.2] by  $\kappa_p(\mathbf{A}) = \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p$ .

The condition numbers  $\kappa_p(\mathbf{A})$  of a nonsingular matrix  $\mathbf{A}$  for  $p = 1, \infty$  can be easily computed and give a good measure of conditioning in CAGD applications [7,20]. We will use them to study the conditioning of algorithms for degree elevation of Bernstein polynomial representation. To determine  $\|[\mathbf{\Omega}_\mu^{n,k}]^{-1}\|_p$  for  $\mu = 0, \dots, n$  and  $p = 1, \infty$  we consider a generalization of a well-known fact.

**Proposition 4.2.** Let  $\mathbf{\Omega}_\mu^{n,k}$ ,  $0 \leq \mu \leq n$ , be given by (3)–(4) and let  $\mathbf{\Upsilon}_\mu^{n,k} = \{v_{i,j}^{n,k}\}_{i,j=0}^{\mu,\mu}$  be its inverse. Then

$$v_{i,j}^{n,k} = \begin{cases} (-1)^{i-j} \binom{i-j+k-1}{k-1} \binom{n+k}{j} / \binom{n}{i}, & j \leq i, k \neq 0, \\ 1, & j = i, k = 0, \\ 0, & \text{otherwise.} \end{cases}$$

**Proof.** It is known that  $\mathbf{\Upsilon}_n^{n,k} = [\mathbf{\Omega}_n^{n,k}]^{-1}$  (see, e.g., [18, Section 3.2]), so the result follows, since the matrix  $\mathbf{\Omega}_\mu^{n,k}$  is for  $\mu = 0, \dots, n$  triangular.  $\square$

From (3) and Proposition 4.2, by the well-known formulae (see, e.g., [27, Section 6.2]) for  $\|\mathbf{A}\|_p$  with  $p = 1, \infty$ ,  $\mathbf{A} \in \mathbb{R}^{(\mu+1) \times (\mu+1)}$ ,  $\mu = 0, \dots, n$ , we obtain

$$\|\mathbf{\Omega}_\mu^{n,k}\|_1 = \max_{0 \leq j \leq \mu} \left\{ \sum_{i=j}^{\mu} \omega_{i,j}^{n,k} \right\}, \quad \|\mathbf{\Upsilon}_\mu^{n,k}\|_1 = \max_{0 \leq j \leq \mu} \left\{ \sum_{i=j}^{\mu} |v_{i,j}^{n,k}| \right\}, \quad (15)$$

$$\|\mathbf{\Omega}_\mu^{n,k}\|_\infty = \max_{0 \leq i \leq \mu} \left\{ \sum_{j=0}^i \omega_{i,j}^{n,k} \right\}, \quad \|\mathbf{\Upsilon}_\mu^{n,k}\|_\infty = \max_{0 \leq i \leq \mu} \left\{ \sum_{j=0}^i |v_{i,j}^{n,k}| \right\}, \quad (16)$$

$$\kappa_p(\mathbf{\Omega}_\mu^{n,k}) = \|\mathbf{\Omega}_\mu^{n,k}\|_p \|\mathbf{\Upsilon}_\mu^{n,k}\|_p, \quad p = 1, \infty. \quad (17)$$

To evaluate the condition numbers  $\kappa_p(\mathbf{\Omega}_\mu^{n,k})$  for  $\mu = 0, \dots, n$ ,  $p = 1, \infty$  we establish now four auxiliary relations:

$$\sum_{i=0}^{n+k} \omega_{i,j}^{n,k} = \sum_{i=j}^{j+k} \omega_{i,j}^{n,k} = 1 + \frac{k}{n+1} \quad \text{for } j = 0, \dots, n, \quad (18)$$

$$\sum_{j=0}^n \omega_{i,j}^{n,k} = \sum_{j=\max(0,i-k)}^{\min(n,i)} \omega_{i,j}^{n,k} = 1 \quad \text{for } i = 0, \dots, n+k, \quad (19)$$

$$\sum_{j=0}^{i+1} |v_{i+1,j}^{n,k}| > \sum_{j=0}^i |v_{i,j}^{n,k}| \quad \text{for } 1 \leq k \leq n+1, \quad 0 \leq i \leq n-1, \quad (20)$$

$$|v_{i,j}^{n,l}| > |v_{i,j}^{n,k}| \quad \text{for } l > k, \quad i, j = 0, \dots, n. \quad (21)$$

Note that (18) is a generalization of the well-known equality [26, (4.6)]; the former yields the latter for  $n = 2j$ . We prove (18) by induction on  $k$ . It is clearly true for  $k = 0$ . Assume it is true for  $k$  (and all  $n \in \mathbb{N}$ ,  $j = 0, \dots, n$ ). Then, using (4) and  $\binom{k+1}{i-j} = \binom{k}{i-j} + \binom{k}{i-j-1}$  for  $i = j+1, \dots, j+k$ , we can write

$$\begin{aligned} \sum_{i=0}^{n+k+1} \omega_{i,j}^{n,k+1} &= \binom{n}{j} \left\{ \sum_{i=j}^{j+k} \binom{k}{i-j} / \binom{n+k+1}{i} + \sum_{i=j+1}^{j+k+1} \binom{k}{i-j} / \binom{n+k+1}{i} \right\} \\ &= \binom{n}{j} \left\{ \left[ \sum_{i=j}^{j+k} \omega_{i,j}^{n+1,k} \right] / \binom{n+1}{j} + \left[ \sum_{i=j+1}^{j+1+k} \omega_{i,j}^{n+1,k} \right] / \binom{n+1}{j+1} \right\} \\ &= \binom{n}{j} (n+k+2) \left\{ \binom{n+1}{j}^{-1} + \binom{n+1}{j+1}^{-1} \right\} / (n+2) \\ &= 1 + (k+1)/(n+1), \end{aligned}$$

which completes the proof of (18). Equality (19) is equivalent to the well-known equality [26, (3.4)] and follows from the relation:

$$\sum_{s=0}^k \binom{k}{s} x^s \sum_{j=0}^n \binom{n}{j} x^j = \sum_{i=0}^{n+k} \binom{n+k}{i} x^i \quad \text{for all } x \in \mathbb{R}.$$

To prove (20) it suffices to observe that

$$\begin{aligned} \sum_{j=0}^{i+1} |v_{i+1,j}^{n,k}| &= \frac{i+1}{n-i} \left[ \binom{i+k}{k-1} + \sum_{j=1}^{i+1} \binom{i-j+k}{k-1} \binom{n+k}{j} \right] / \binom{n}{i} \\ &= \frac{i+1}{n-i} \binom{i+k}{k-1} / \binom{n}{i} + \sum_{j=0}^i \frac{n+k-j}{n-i} \frac{i+1}{j+1} |v_{i,j}^{n,k}| \end{aligned}$$

and that

$$\frac{n+k-j}{n-i} \frac{i+1}{j+1} \geq 1 \quad \text{for all } j = 0, \dots, i.$$

Relation (21) is easily verified.

By (15)–(20), for  $k \geq 1$  we have

$$\kappa_1(\Omega_\mu^{n,k}) = \begin{cases} \max_{0 \leq j \leq \mu} \left\{ \sum_{i=0}^{\mu} \omega_{i,j}^{n,k} \right\} \max_{0 \leq j \leq \mu} \left\{ \sum_{i=j}^{\mu} |v_{i,j}^{n,k}| \right\}, & \mu = 0, \dots, k-1, \\ \left( 1 + \frac{k}{n+1} \right) \max_{0 \leq j \leq \mu} \left\{ \sum_{i=j}^{\mu} |v_{i,j}^{n,k}| \right\}, & \mu = k, \dots, n, \end{cases} \quad (22)$$

$$\kappa_\infty(\Omega_\mu^{n,k}) = \|\Upsilon_\mu^{n,k}\|_\infty = \sum_{j=0}^{\mu} |v_{\mu,j}^{n,k}|, \quad \mu = 0, \dots, n. \quad (23)$$



We establish now conditioning measures for the singular matrices  $\mathbf{A} \in \mathbb{R}^{(m+1) \times (n+1)}$ , where  $m > n$ , as a generalization of those usually used for the nonsingular matrices.

**Definition 4.3.** Let  $\mathbf{A} \in \mathbb{R}^{(m+1) \times (n+1)}$ ,  $m \geq n$ . Denote by  $\mathbf{A}[i_0, \dots, i_n]$ ,  $0 \leq i_0 < \dots < i_n \leq m$  the submatrix of  $\mathbf{A}$  formed by the intersection of the rows  $i_0, \dots, i_n$  and suppose there exist  $0 \leq i_0 < \dots < i_n \leq m$  such that  $\det(\mathbf{A}[i_0, \dots, i_n]) \neq 0$ . We define

$$\kappa_p(\mathbf{A}) = \max_{0 \leq i_0 < \dots < i_n \leq m} \{\kappa_p(\mathbf{A}[i_0, \dots, i_n]) : \det(\mathbf{A}[i_0, \dots, i_n]) \neq 0\}, \quad p = 1, \infty.$$

The following consequence of (15)–(17) and of Definition 4.3 is noteworthy.

**Corollary 4.4.**

$$\kappa_p(\mathbf{\Omega}_\mu^{n,k}) \leq \kappa_p(\mathbf{\Omega}_v^{n,k}) \quad \text{for all } \mu \leq v, \quad \mu, v = 0, \dots, n+k, \quad p = 1, \infty.$$

In Section 5, we will use the condition numbers of the matrices  $\mathbf{\Omega}_\mu^{n,k}$ , where  $\mu = 0, \dots, n+k$ , to determine the conditioning of algorithms for degree elevation of Bernstein polynomial representation.

## 5. Conditioning of algorithms

### 5.1. Preliminary definitions and results

The algorithm  $\mathbf{a}_\mu^{n,k}$  yields vectors (14) independently of each other. These vectors are determined by two linear transformations whose conditionings are expressed for a given  $p$  by  $\kappa_p(\mathbf{\Omega}_\mu^{n,k})$ ,  $\kappa_p(\mathbf{\Omega}_{n+k-\mu-1}^{n,k})$ , respectively. This leads us to the following definition.

**Definition 5.1.** We set  $\kappa_p(\mathbf{\Omega}_{-1}^{n,k}) = 0$  and define the *condition number of the algorithm  $\mathbf{a}_\mu^{n,k}$*  by

$$\kappa_p(\mathbf{a}_\mu^{n,k}) = \max\{\kappa_p(\mathbf{\Omega}_\mu^{n,k}), \kappa_p(\mathbf{\Omega}_{n+k-\mu-1}^{n,k})\} \quad \text{for } p = 1, \infty.$$

The above-defined condition number of an algorithm  $\mathbf{a}_\mu^{n,k}$  is, for given  $n$  and  $k$ , uniquely determined by the value of  $\mu = -1, \dots, n+k$ . This property and the symmetry of the function  $f(x) = \kappa_p(\mathbf{a}_{x+(n+k-1)/2}^{n,k})$  are direct consequences of Definition 5.1. Moreover, the function  $f(x)$  is for  $x \geq 0$  nondecreasing. We express these facts in the form of a proposition and a theorem.

**Proposition 5.2.** For any  $\mathbf{a}_\mu^{n,k}$ ,  $\mathbf{b}_\mu^{n,k}$  we have

$$\kappa_p(\mathbf{a}_\mu^{n,k}) = \kappa_p(\mathbf{b}_\mu^{n,k}) = \kappa_p(\mathbf{a}_{n+k-\mu-1}^{n,k}), \quad p = 1, \infty.$$

**Theorem 5.3.** Let  $\mu, v = -1, \dots, n+k$ ,  $p = 1, \infty$ .

$$\text{If } |\mu - (n+k-1)/2| \leq |v - (n+k-1)/2| \quad \text{then} \quad \kappa_p(\mathbf{a}_\mu^{n,k}) \leq \kappa_p(\mathbf{a}_v^{n,k}).$$

**Proof.** We define

$$x = \mu - (n+k-1)/2, \quad y = v - (n+k-1)/2, \quad f(x) = \kappa_p(\mathbf{a}_{x+(n+k-1)/2}^{n,k}).$$

If  $0 \leq x \leq y$  then it easily verified that

$$n + k - 1 - v \leq n + k - 1 - \mu \leq \mu \leq v,$$

thus by Definition 5.1 and Corollary 4.4 we obtain

$$f(x) = \kappa_p(\mathbf{a}_\mu^{n,k}) \leq \kappa_p(\mathbf{a}_v^{n,k}) = f(y).$$

From Proposition 5.2 follows that  $f(x) = f(-x)$ , which completes the thesis.  $\square$

Using Definition 5.1, Proposition 5.2, (22)–(23) and some relations proved in [37] (see [37, (27)–(28)] with  $\Omega_\mu^{n,1} = \Omega_\mu^{n+1}$ ,  $\Upsilon_\mu^{n,1} = \Upsilon_\mu^{n+1}$ ), we obtain

$$\kappa_\infty(\mathbf{a}_{\lfloor n/2 \rfloor}^{n,1}) = \kappa_\infty(\mathbf{a}_{\lfloor (n+1)/2 \rfloor}^{n,1}) = 2^n \left/ \binom{n}{\lfloor n/2 \rfloor} + ((-1)^{n+1} - 1)/2 \right. < n + 1, \quad (24)$$

$$\kappa_\infty(\mathbf{b}_n^{n,1}) = \kappa_\infty(\mathbf{b}_0^{n,1}) = 2^{n+1} - 1, \quad (25)$$

$$\kappa_1(\mathbf{a}_{\lfloor n/2 \rfloor}^{n,1}) = \kappa_1(\mathbf{a}_{\lfloor (n+1)/2 \rfloor}^{n,1}) < \lfloor n/2 \rfloor (n + 2), \quad (26)$$

$$\kappa_1(\mathbf{b}_n^{n,1}) = \kappa_1(\mathbf{b}_0^{n,1}) = \left(1 + \frac{1}{n+1}\right) \binom{n+1}{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=\lfloor \frac{n+1}{2} \rfloor}^n \binom{n}{i}^{-1} > \frac{2^{n+1}}{n+1} \quad (27)$$

for any algorithm  $\mathbf{a}_{\lfloor n/2 \rfloor}^{n,1}$  and  $\mathbf{b}_n^{n,1}$ , where  $\lfloor \cdot \rfloor$  is the floor function.

## 5.2. Nonsingular and singular algorithms

Definition 2.3 and Proposition 4.1 motivate us to establish two classes of the algorithms for degree elevation of Bernstein polynomial representation.

**Definition 5.4.** The algorithm  $\mathbf{a}_\mu^{n,k}$  is nonsingular if  $\mu \in [k-1, n]$ , otherwise it is said to be singular.

Using the above definition we can easily state two facts.

### Properties 5.5.

1.  $\mathbf{b}_\mu^{n,n+1}$  is nonsingular if and only if  $\mu = n$ .
2. All the current algorithms for degree elevation of Bernstein polynomial representations, that is,  $\mathbf{s}_{n+k}^{n,k}$ ,  $\mathbf{s}_{n+k}^{n,k}$  and  $\mathbf{t}_{n+k}^{n,k}$ , are singular.

We yield now a characterization of the condition numbers of the nonsingular and singular algorithms.

**Theorem 5.6.** Let  $p = 1, \infty$ . If  $\mathbf{c}_\mu^{n,k}$  is nonsingular then

$$\kappa_p(\mathbf{a}_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}) \leq \kappa_p(\mathbf{c}_\mu^{n,k}) \leq \kappa_p(\mathbf{b}_n^{n,k}) = \kappa_p(\Omega_n^{n,k}) \quad (28)$$

and  $\mathbf{a}_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}, \mathbf{b}_n^{n,k}$  are nonsingular. If  $\mathbf{d}_v^{n,k}$  is singular then for any  $\mathbf{b}_n^{n,k}$

$$\kappa_p(\mathbf{\Omega}_n^{n,k}) \leq \kappa_p(\mathbf{b}_n^{n,k}) \leq \kappa_p(\mathbf{d}_v^{n,k}), \quad (29)$$

moreover, if  $k > n + 1$  then

$$\kappa_p(\mathbf{b}_n^{n,n+1}) < \kappa_p(\mathbf{d}_v^{n,k}). \quad (30)$$

**Proof.** It is easily verified, using Definition 5.4, that if  $\mu \in [k-1, n] \neq \emptyset$  then any  $\mathbf{a}_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$  is nonsingular (for  $\mathbf{b}_n^{n,k}$  see Property 5.5.1). We set  $\tau = \lfloor (n+k-1)/2 \rfloor$  and note that if  $\mu \in [k-1, n]$  then either

$$k \leq \mu \leq \tau, \quad \text{and then } \tau \leq n+k-\tau-1 \leq n+k-\mu-1 \leq n$$

or

$$\tau < \mu \leq n, \quad \text{and then } n+k-\mu-1 \leq \tau \leq n+k-\tau-1 = \lfloor (n+k)/2 \rfloor \leq \mu.$$

Thus, using Corollary 4.4 and Definition 5.1, we obtain (28). Similarly, if  $\mu \notin [k-1, n]$  then, using Proposition 4.1, Corollary 4.4 and Definition 5.1, we obtain (29). If  $k > n + 1$  then  $\min\{v, n+k-v-1\} = (n+k-1)/2 > n$ , thus by Definition 5.1 and Corollary 4.4 we can write

$$\kappa_p(\mathbf{b}_n^{n,n+1}) = \kappa_p(\mathbf{\Omega}_n^{n,n+1}), \quad \kappa_p(\mathbf{\Omega}_n^{n,k}) \leq \kappa_p(\mathbf{d}_v^{n,k}), \quad p = 1, \infty$$

and, by (21)–(23), we have

$$\kappa_p(\mathbf{\Omega}_n^{n,n+1}) < \kappa_p(\mathbf{\Omega}_n^{n,k}), \quad p = 1, \infty,$$

which gives (30).  $\square$

Theorem 5.6 together with Definition 5.4 enables us to obtain the following relations for the onefold degree elevation algorithms for Bernstein polynomial representation.

### Corollary 5.7.

$$\kappa_p(\mathbf{a}_{\lfloor \frac{n}{2} \rfloor}^{n,1}) \leq \kappa_p(\mathbf{c}_\mu^{n,1}) \leq \kappa_p(\mathbf{b}_n^{n,1}) \leq \kappa_p(\mathbf{d}_{n+1}^{n,1}) \quad \text{for } \mu = 0, \dots, n, \quad p = 1, \infty.$$

To order the condition numbers of the algorithms with the same values of  $n, \mu$  and different degree elevations we prove a theorem.

**Theorem 5.8.** For any nonsingular  $\mathbf{c}_\mu^{n,k}, \mathbf{c}_\mu^{n,k+1}$  we have

$$\kappa_p(\mathbf{c}_\mu^{n,k}) < \kappa_p(\mathbf{c}_\mu^{n,k+1}), \quad p = 1, \infty.$$

**Proof.** Setting  $m = \max\{\mu, n+k-\mu-1\}$ ,  $M = \max\{\mu, n+k-\mu\}$ , we have  $m \leq M$  and by Definition 5.1 together with Corollary (4.4) we obtain

$$\kappa_p(\mathbf{c}_\mu^{n,k}) = \kappa_p(\mathbf{\Omega}_m^{n,k}), \quad \kappa_p(\mathbf{\Omega}_m^{n,k+1}) \leq \kappa_p(\mathbf{c}_\mu^{n,k+1}) = \kappa_p(\mathbf{\Omega}_M^{n,k+1}). \quad (31)$$

Table 3

The condition numbers  $\kappa_p(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor})$ ,  $\kappa_p(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor})$ ,  $p = 1, \infty$ , where  $A(n) = \lfloor (n + \lfloor (n+1)/2 \rfloor - 1)/2 \rfloor$ , rounded to the first two decimal digits

$n$	1	2	3	4	5	6	7
$\kappa_1(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor})$	3.0E + 0	2.0E + 0	7.5E + 0	1.4E + 1	6.7E + 1	3.2E + 1	1.5E + 2
$\kappa_\infty(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor})$	3.0E + 0	2.0E + 0	7.7E + 0	1.8E + 1	1.0E + 2	4.7E + 1	2.4E + 2
$\kappa_1(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor})$	3.0E + 0	6.0E + 0	3.5E + 1	7.7E + 1	5.6E + 2	1.3E + 3	1.1E + 4
$\kappa_\infty(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor})$	3.0E + 0	7.0E + 0	4.9E + 1	1.3E + 2	1.0E + 3	2.8E + 3	2.3E + 4
$n$	8	9	10	11	12	13	14
$\kappa_1(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor})$	3.2E + 2	1.8E + 3	7.7E + 2	3.9E + 3	9.0E + 3	5.1E + 4	2.1E + 4
$\kappa_\infty(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor})$	5.9E + 2	3.4E + 3	1.4E + 3	7.9E + 3	2.0E + 4	1.2E + 5	4.7E + 4
$\kappa_1(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor})$	2.7E + 4	2.2E + 5	5.9E + 5	4.9E + 6	1.3E + 7	1.1E + 8	3.0E + 8
$\kappa_\infty(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor})$	6.6E + 4	5.5E + 5	1.6E + 6	1.4E + 7	3.9E + 7	3.4E + 8	9.7E + 8

Since  $\mathbf{c}_\mu^{n,k}$ ,  $\mathbf{c}_\mu^{n,k+1}$  are nonsingular we have  $k, M \leq n$  and we can easily verify, using Proposition 4.2, that

$$|v_{i,j}^{n,k}| < |v_{i,j}^{n,k+1}| \quad \text{for } j \leq i, \quad i, j = 0, \dots, m,$$

and using (4), (18), that

$$\sum_{i=0}^k \omega_{i,0}^{n,k+1} = 1 + \frac{k+1}{n+1} - \frac{1}{\binom{n+k+1}{k+1}} > 1 + \frac{k}{n+1} \quad \text{for } 1 \leq k \leq n.$$

Thus by (22) and (23) we obtain  $\kappa_p(\Omega_m^{n,k}) < \kappa_p(\Omega_m^{n,k+1})$ ,  $p = 1, \infty$ . This together with (31) completes the proof.  $\square$

### 5.3. Numerical results

It seemed impossible to find closed-form expressions for the condition numbers of the nonsingular algorithms  $\mathbf{c}_\mu^{n,k}$  with  $k > 1$ . To estimate the size of these condition numbers we will use the first part of Theorem 5.6.

To do this for the whole range of  $k$  for which the algorithms  $\mathbf{c}_\mu^{n,k}$  are nonsingular, that is, for  $k \in [1, n+1]$  we consider first the endpoints 1,  $n+1$  and the point  $\lfloor (n+1)/2 \rfloor$  of this interval.

For  $k = 1$  the considered condition numbers are known (see (24)–(27)). Using Definition 5.1 together with (22) and (23), we compute these numbers for  $k = \lfloor (n+1)/2 \rfloor$ ,  $n+1$  and  $n = 1, \dots, 256$ . For  $n = 1, \dots, 14$  we list them in Tables 3 and 4, and for  $n = 15, \dots, 256$  we obtain inequalities (32) and (33).

$$(2.3)^{n-N} \kappa_p(\mathbf{a}_{A(N)}^{N, \lfloor (N+1)/2 \rfloor}) < \kappa_p(\mathbf{a}_{A(n)}^{n, \lfloor (n+1)/2 \rfloor}) < (2.5)^{n-N} \kappa_p(\mathbf{a}_{A(N)}^{N, \lfloor (N+1)/2 \rfloor}),$$

$$(4.8)^{n-N} \kappa_p(\mathbf{b}_N^{N, \lfloor (N+1)/2 \rfloor}) < \kappa_p(\mathbf{b}_n^{n, \lfloor (n+1)/2 \rfloor}) < (5.2)^{n-N} \kappa_p(\mathbf{b}_N^{N, \lfloor (N+1)/2 \rfloor}), \quad (32)$$

$$(7.0)^{n-N} \kappa_p(\mathbf{b}_N^{N, N+1}) < \kappa_p(\mathbf{b}_n^{n, n+1}) < (8.1)^{n-N} \kappa_p(\mathbf{b}_N^{N, N+1}), \quad (33)$$

Table 4

The condition numbers  $\kappa_p(\mathbf{b}_n^{n,n+1})$ ,  $p = 1, \infty$ , rounded to the first two decimal digits

$n$	1	2	3	4	5	6	7
$\kappa_1(\mathbf{b}_n^{n,n+1})$	5.0E + 0	3.3E + 1	1.8E + 2	1.2E + 3	7.4E + 3	5.1E + 4	3.4E + 5
$\kappa_\infty(\mathbf{b}_n^{n,n+1})$	5.0E + 0	3.1E + 1	2.1E + 2	1.5E + 3	1.1E + 4	7.8E + 4	5.8E + 5
$n$	8	9	10	11	12	13	14
$\kappa_1(\mathbf{b}_n^{n,n+1})$	2.5E + 6	1.7E + 7	1.3E + 8	9.1E + 8	6.8E + 9	4.9E + 10	3.7E + 11
$\kappa_\infty(\mathbf{b}_n^{n,n+1})$	4.4E + 6	3.3E + 7	2.5E + 8	1.9E + 9	1.5E + 10	1.1E + 11	8.7E + 11

where  $p = 1, \infty$  and  $A(n)$ ,  $N$  are given by

$$A(n) = \lfloor (n + \lfloor (n+1)/2 \rfloor - 1)/2 \rfloor, \quad N = (n + 1 \bmod 4) + 11.$$

Notice that  $N = 11, 12, 13, 14$ , thus for every  $n > 14$  the values of

$$\kappa_p(\mathbf{a}_{A(N)}^{N, \lfloor (N+1)/2 \rfloor}), \quad \kappa_p(\mathbf{b}_N^{N, \lfloor (N+1)/2 \rfloor}), \quad \kappa_p(\mathbf{b}_N^{N, N+1})$$

are for  $p = 1, \infty$  listed in one of the last four columns of [Tables 3 and 4](#).

Thus (24)–(27) together with Theorem 5.6, [Tables 3 and 4](#) and (32)–(33) enable us to estimate the condition numbers  $\kappa_p(\mathbf{c})$ , where  $p = 1, \infty$ , of all the degree elevation algorithms  $\mathbf{c}$  having the form

$$\mathbf{c}_\mu^{n,1}, \quad \mathbf{c}_\mu^{[(n+1)/2]}, \quad \mathbf{c}_\mu^{n,n+1} \quad \text{with } n = 1, \dots, 256. \quad (34)$$

To estimate the condition numbers of the nonsingular degree elevation algorithms  $\mathbf{c}_\mu^{n,k}$  for  $k \neq 1$ ,  $\lfloor (n+1)/2 \rfloor, n+1$  and  $n = 1, \dots, 256$  one can use the estimates of the condition numbers of the algorithms considered in (34) together with Theorem 5.8.

To estimate the condition numbers of the singular degree elevation algorithms  $\mathbf{d}_\mu^{n,k}$  for  $n = 1, \dots, 256$  one can use (29) together with [Table 4](#) and (33).

## 6. Summary and conclusions

Talbot [38] proposed the algorithm  $\mathbf{s}_{n+1}^{n,1}$  in 1971 (see also [25, Section 6]). This algorithm is explicitly cited or used in [1, Section 10.1; 3, Section 2.3.1.3; 4, Section 4.1.3; 5, Section 9; 9, Section 3; 12, (7)–(8); 14, Section 8; 15, Section 4.5; 16, Section 51; 17, Section 1.3 (d); 18, Section 3.2; 22, Section 1; 23, Section 1; 28, Section 4.1.1; 29, (6); 30, Section 4; 31, Section 3.3; 32, Section 2.1; 33, p. 21; 36, Section 5–9; 39, Section 1; 41, Section 5.1.8]. The authors who cite the algorithm  $\mathbf{s}_{n+1}^{n,1}$  also usually consider the algorithm  $\mathbf{s}_{n+k}^{n,k}$ .

The algorithm  $\mathbf{s}_{n+k}^{n,k}$  was first cited in [18, Section 3.2]. It is considered or used in [2, Section 3.1; 21, Section 3; 23, Section 2; 24, Section 2; 28, Section 4.1.1; 30, Section 4; 33, p. 21]. Trump and Prautzsch established the algorithm  $\mathbf{t}_{n+k}^{n,k}$  in [39].

The costs of these algorithms can be, using [Tables 1 and 2](#), characterized as follows.

### Conclusion 6.1.

- The costs of the algorithms  $s_\mu^{n,k}$  do not depend on  $\mu$ .
- The algorithm  $s_\mu^{n,k}$  is not more and for  $k \geq 2$  less expensive than the algorithm  $t_{n+k}^{n,k}$ .
- The algorithm  $t_{n+k}^{n,k}$  is for  $k \geq 2$  less expensive than the algorithms  $\tilde{t}_{n+k}^{n,k}$  (see Algorithms 2.5) and  $\bar{s}_{n+k}^{n,k}$ .

The conditioning of the degree elevation of Bernstein polynomial representation was first considered by Farouki and Rajan (see [18, Section 2.5]) in the context of the application (d) of Section 1.

We showed in Section 5.1 that the condition number of any degree elevation algorithm  $a_\mu^{n,k}$  is for given  $n$  and  $k$  uniquely determined by the value of  $\mu$  (see Proposition 5.2). This, together with Corollary 5.7 and with (24)–(27), leads us to the following conclusion for the onefold degree elevation algorithms.

**Conclusion 6.2.** For high  $n$ , any algorithm  $a_{\lfloor n/2 \rfloor}^{n,1}$  is substantially better conditioned than any algorithm  $d_{n+1}^{n,1}$ . In particular, the algorithm  $s_{\lfloor n/2 \rfloor}^{n,1}$  is substantially better conditioned than the current degree elevation algorithms  $s_{n+1}^{n,1}$  and  $t_{n+1}^{n,1}$ .

It is worth noting that all the current algorithms for degree elevation of Bernstein polynomial representation, that is, the algorithms  $s_{n+k}^{n,k}$ ,  $\bar{s}_{n+k}^{n,k}$ ,  $t_{n+k}^{n,k}$ , are singular (see Property 5.5.2) and that any singular algorithm  $d_v^{n,k}$  is not better conditioned than any nonsingular algorithm  $c_\mu^{n,k}$  (see Theorem 5.6).

If an algorithm  $d_v^{n,k}$  is singular then by Definition 5.4

$$\text{either } k \leq n+1 \text{ and } v \notin [k-1, n] \text{ or } k > n+1.$$

In the first case, any algorithm  $a_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$  is nonsingular, and for  $k \leq \lfloor (n+1)/2 \rfloor$  it is substantially better conditioned than the algorithm  $d_v^{n,k}$  (see Theorem 5.6 together with (24)–(27), Tables 3 and 4, (32)–(33) and Theorem 5.8) which should be replaced by a degree elevation algorithm of the  $a_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$  type.

For  $k = \lfloor (n+1)/2 \rfloor$ , however, the condition numbers even of the  $a_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$  type algorithms are characterized by a significant exponential growth with the degree  $n \in [1, 256]$  (see Table 3 and (32)). For  $k$  satisfying  $\lfloor (n+1)/2 \rfloor < k \leq n+1$ , this growth is by Theorems 5.6 and 5.8 still faster.

In the second case, the algorithm  $d_v^{n,k}$  is worse conditioned than any algorithm  $b_n^{n,n+1}$  (see Theorem 5.6) and the condition numbers for  $b_n^{n,n+1}$  increase exponentially dramatically fast with  $n \in [1, 256]$  (see Table 4 and (33)). They increase faster than the corresponding condition numbers of the Bernstein-power basis transformation, which are approximately equal to  $3^n \sqrt{n+1}$  (see [20, Section 6]).

On the other hand we note, that up to now there is just one well-known algorithm of the type  $a_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$ , this is the algorithm  $s_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$ .

These facts and the well-established in CAGD usefulness estimation of algorithms in function of their condition number growth (see, e.g., [21, Section 4.2]) lead us to the following conclusion for the  $k$ -fold degree elevation of Bernstein polynomial representation with  $n \in [1, 256]$ , that is, within the range of  $n$  covering the practical applications of these algorithms.

**Conclusion 6.3.**

For  $k \leq n + 1$ , the algorithm  $\mathbf{s}_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$  is not worse conditioned than any algorithm  $\mathbf{a}_{\mu}^{n,k}$  for degree elevation of Bernstein polynomial representation,  
 for  $k \leq \lfloor (n + 1)/2 \rfloor$ , this algorithm is substantially better conditioned than the current algorithms  $\mathbf{s}_{n+k}^{n,k}, \mathbf{t}_{n+k}^{n,k}$ .  
 For  $\lfloor (n + 1)/2 \rfloor \leq k \leq n + 1$ , the  $k$ -fold degree elevation of Bernstein polynomial representation should be avoided except for very small  $n$ ,  
 for  $k > n + 1$ , this degree elevation seems to be computationally meaningless.

For the usual applications of degree elevation, that is (see Section 1), for:

- (i) computing the sum, difference, quotient and remainder of two polynomials represented in two Bernstein bases of different degrees,
- (ii) making closer the connection between a curve and its control polygon,
- (iii) increasing the range of possible shape modification of a curve,
- (iv) creating of composite curves or surfaces from curves and surfaces defined by polynomials in the Bernstein bases of different degrees,
- (v) transferring of polynomials between Bernstein bases of different degrees,
- (vi) decreasing the sensitivity of the roots of a polynomial to random perturbations in its coefficients,

we obtain from Conclusion 6.3 the following one.

**Conclusion 6.4.** Assume that the degree  $n$  is elevated to  $m$ , then

for  $m - n < \lfloor (n + 1)/2 \rfloor$ , the algorithm  $\mathbf{s}_{\lfloor \frac{m-1}{2} \rfloor}^{n,m-n}$  guarantees much more accurate results than any other current algorithm for degree elevation of Bernstein polynomial representation,  
 for  $m - n \geq \lfloor (n + 1)/2 \rfloor$ , the result could be very inaccurate except for very small  $n$ .

This in turn

- shows that in the above application (vi) the degree elevation from  $n$  to  $m$  can, especially if  $m - n \geq \lfloor (n + 1)/2 \rfloor$ , cancel the enhanced root conditioning of the degree-elevated basis (cf. [18, Section 2.4]),
- provides supplementary arguments, to that given in [19], against the use of high-degree Bernstein bases in CAGD, and
- argues together with the argument given in [16, Section 5.2] against the idea (see, e.g., [34, Section 1]) to display the control polygons of curves instead of the curves themselves.

We listed here, in Algorithms 2.4 and 2.5, the current and some new algorithms for degree elevation of Bernstein polynomial representation. Our list is, of course, nonexhaustive. Other algorithms can be established for degree elevation of Bernstein polynomial representation, but such an algorithm is only then worth to be considered if it is either cheaper or better conditioned than the best here mentioned one, that is (see Conclusions 6.1–6.3), the algorithm  $\mathbf{s}_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$ . The algorithm  $\bar{\mathbf{t}}_{n+k}^{n,k}$ , for example, that is a direct derivative of the algorithm  $\mathbf{t}_{n+k}^{n,k}$ , is included in this paper only for the sake of completeness, since

although only slightly more expensive than  $t_{n+k}^{n,k}$ , it is, as the latter, substantially worse conditioned and more expensive than the algorithm  $s_{\lfloor \frac{n+k-1}{2} \rfloor}^{n,k}$ .

## References

- [1] R.H. Bartels, J.C. Beatty, B.A. Barsky, An Introduction to Splines for use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers Inc., Los Altos, 1987.
- [2] J. Berchtold, A. Bowyer, Robust arithmetic for multivariate Bernstein-form polynomials, *Comput. Aided Des.* 32 (2000) 681–689.
- [3] P. Bézier, Essai de définition numérique des courbes et des surfaces expérimentales, Doctor's Thesis. University Paris VI, 1977.
- [4] P. Bézier, The Mathematical Basis of the UNISURF CAD System, Butterworths, London, 1986.
- [5] W. Böhm, G. Farin, J. Kahmann, A survey of curve and surface methods in CAGD, *Comput. Aided Geom. Des.* 1 (1984) 1–60.
- [6] E. Cohen, L.L. Schumaker, Rates of convergence of control polygons, *Comput. Aided Geom. Des.* 2 (1985) 229–235.
- [7] M. Daniel, J.C. Daubisse, The numerical problem of using Bézier curves and surfaces in the power basis, *Comput. Aided Geom. Des.* 6 (1989) 121–128.
- [8] C. De Boor, B-form basics, in: G. Farin (Ed.), *Geometric Modeling: Algorithms and New Trends*, SIAM, Philadelphia, 1987, pp. 131–148.
- [9] W.A. Denker, G.J. Herron, Generalizing rational degree elevation, *Comput. Aided Geom. Des.* 14 (1997) 399–406.
- [10] G. Farin, Konstruktion und Eigenschaften von Bézier-Kurven und -Flächen, Master's Thesis, Technical University Braunschweig, FRG 1977.
- [11] G. Farin, Subsplines über Dreiecken, Doctor's Thesis, Technical University Braunschweig, FRG, 1979.
- [12] G. Farin, Algorithms for rational Bézier curves, *Comput. Aided Des.* 15 (1983) 73–77.
- [13] G. Farin, Triangular Bernstein–Bézier patches, *Comput. Aided Geom. Des.* 3 (1986) 83–127.
- [14] G. Farin, Rational curves and surfaces, in: T. Lyche, L.L. Schumaker (Eds.), *The Mathematical Methods in Computer Aided Geometric Design*, Academic Press Inc., New York, 1989, pp. 215–238.
- [15] G. Farin, A. Worsey, Reparametrization and degree elevation of rational Bézier curves, in: G. Farin (Ed.), *NURBS for Curve and Surface Design*, SIAM, Philadelphia, 1991, pp. 47–58.
- [16] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic Press, New York, 1993.
- [17] R.T. Farouki, V.T. Rajan, On the numerical condition of polynomials in Bernstein form, *Comput. Aided Geom. Des.* 4 (1987) 191–216.
- [18] R.T. Farouki, V.T. Rajan, Algorithms for polynomials in Bernstein form, *Comput. Aided Geom. Des.* 5 (1988) 1–26.
- [19] R.T. Farouki, C.A. Neff, On the numerical condition of Bernstein–Bézier subdivision process, *Math. Comput.* 55 (1990) 637–647.
- [20] R.T. Farouki, On the stability of transformations between power and Bernstein polynomial forms, *Comput. Aided Geom. Des.* 8 (1991) 29–36.
- [21] R.T. Farouki, Legendre–Bernstein basis transformation, *J. Comput. Appl. Math.* 119 (1–2) (2000) 145–160.
- [22] J.C. Fiorot, P. Jeannin, A blossoming approach to accuracy of the degree elevation process, *C. R. Acad. Sci. Paris Ser.* 1 322 (1996) 593–598.
- [23] J.C. Fiorot, P. Jeannin, A blossoming approach to accuracy of the degree elevation process, *Numer. Algorithms* 13 (1996) 265–306.
- [24] M.S. Floater, T. Lyche, Asymptotic convergence of degree-raising, *Adv. Comput. Math.* 12 (2000) 175–187.
- [25] A.R. Forrest, Interactive interpolation and approximation by Bézier polynomials, *Comput. J.* 15 (1972) 71–79.
- [26] H.W. Gould, *Combinatorial Identities: A Standardized Set of Tables Listing 500 Binomial Coefficient Summations*, Morgantown, W. Va., 1972.
- [27] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.
- [28] J. Hoschek, D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, B.G. Teubner, Stuttgart, 1989.



- [29] L.M. Kocić, Modification of Bézier curves and surfaces by degree-elevation technique, *Comput. Aided Des.* 23 (1991) 692–699.
- [30] B.-G. Lee, Y. Park, J. Yoo, Application of Legendre–Bernstein basis transformations to degree elevation and degree reduction, *Comput. Aided Geom. Des.* 19 (2002) 709–718.
- [31] R. Martin, H. Shou, I. Voiculescu, A. Bowyer, G. Wang, Comparison of interval methods for plotting algebraic curves, *Comput. Aided Geom. Des.* 19 (2002) 553–587.
- [32] G. Morin, R. Goldman, On the smooth convergence of subdivision and degree elevation for Bézier curves, *Comput. Aided Geom. Des.* 18 (2001) 657–666.
- [33] L. Piegl, W. Tiller, Software-engineering approach to degree elevation of  $\beta$ -spline curves, *Comput. Aided Des.* 26 (1994) 17–28.
- [34] H. Prautzsch, L. Kobbelt, Convergence of subdivision and degree elevation, *Adv. Comput. Math.* 2 (1994) 143–154.
- [35] A. Rababah, Distance for degree raising and reduction of triangular Bézier surfaces, *J. Comput. Appl. Math.* 158 (2003) 233–241.
- [36] D.F. Rogers, J.A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill Publishing Company, New York, 1990.
- [37] B. Szafnicki, A unified approach for degree reduction of polynomials in the Bernstein basis Part I: Real polynomials, *J. Comput. Appl. Math.* 142 (2002) 287–312.
- [38] J.E. Talbot, Experiments towards graphical design of motor bodies, Cambridge University CAD Group Document No. 56, 1971.
- [39] W. Trump, H. Prautzsch, Arbitrarily high degree elevation of Bézier representations, *Comput. Aided Geom. Des.* 13 (1996) 387–398.
- [40] J.H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty’s Stationery Office, London, 1963.
- [41] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*, Springer, Berlin, Heidelberg, New York, London, Paris, 1988.